

BEST AVAILABLE COPY

W O.E.B. Doc. Lit.
112 5 MARS 1981

53/81

proceedings

SEPTEMBER 23-25, 1980 **fall**
COMPCON 80
TWENTY-FIRST IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE
CAPITAL HILTON HOTEL, WASHINGTON, D.C.

IEEE Catalog No. 80CH1598-2C
Library of Congress No. 80-83217

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Additional copies available from:

IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

BEST AVAILABLE COPY



INTEGRATED DISK CACHE SYSTEM WITH FILE ADAPTIVE CONTROL

T. Tokunaga, Y. Hirai and S. Yamamoto

Nippon Electric Co., Ltd.
Tokyo, Japan

Abstract

This paper presents the Integrated Disk Cache (IDC) system developed for ACOS series large-scale computers. The IDC features a unique file adaptive control for high performance gains and improves disk access time by a factor of up to 10. The IDC can be used to improve the throughput and response time of an I/O bound computer system.

Introduction

One of the major performance bottlenecks of large-scale computer systems is said to be slow access time to disk data, which is associated with disk mechanical motions.

To remove this performance bottleneck, several programming and operational techniques have been used. They include decreasing the number of disk accesses by program modifications, reducing seek time by file reallocation and displacing disk data to fast-access devices. However, these techniques are not only costly but also limited in effectiveness and lack in adaptivity to dynamic changes in system environments.

One effective approach to solving these problems is the use of a disk cache or buffered disk. This is a technique to decrease apparent disk access time by means of automatically storing frequently used data in a high-speed buffer memory, which is provided between disks and main memory.

A number of papers discussed the effectiveness of a disk cache ^{1, 2}, but very little has been reported on the development of a disk cache which has been put to practical use. This paper describes the Integrated Disk Cache (IDC) system which has been developed for NEC's ACOS series large-scale computers. Unlike the known disk cache ³, the IDC is integrated into a central input/output processor and features a unique file adaptive control for performance enhancement. The following sections describe the major design considerations, system configuration, file adaptive control and performance gains for the IDC.

Major Design Considerations

In developing a disk cache system, the

following requirements should be taken into consideration, among others.

1) Performance

In order to achieve high performance, frequently accessed data must be found with high probability in the cache. Since data access characteristics widely vary, depending on files and applications used, a disk cache should be adaptive to these various conditions.

2) Reliability and availability

Since disks contain critical system data, sufficient care should be taken to retain data integrity. This requires not only that a disk cache be highly reliable, but also that it should be able to recover data if a failure occurs.

3) Transparency

The function of a disk cache, differing from a high speed file device, is to improve apparent or effective disk access time. Its existence, therefore, should be as transparent as possible at a user level, at an operating system level and at a hardware level.

It is difficult, however, to meet all of the three requirements to the same degree. From performance and availability viewpoints, strict transparency is not desirable, because file categories have considerable influence on efficient cache operations, as follows:

First, disk files are roughly grouped into random file or sequential file categories. Disk cache performance gains for these two file categories are based upon completely different operations: One involves repeated access to same data. The other involves access to prefetched data. It is difficult, therefore, to obtain high performance for both of these operations using a single control. Second, disk files are also grouped into permanent file or temporary file categories. While a permanent file should always be kept in a disk, a temporary file is allocated and used only during a task or job execution; it is not absolutely necessary for it to be stored on a disk. Therefore, separate handling of these two kinds of files would allow significant performance improvement, yet assuring data availability.

Thus it was decided to limit the IDC transparency to an extent that ACOS operating system sees the existence of a disk cache and can provide it with the information on file categories. The disk cache, based on this information, controls the caching operations adaptively according to each file category. This is called a file adaptive control in this paper.

System Configuration and Basic Operations

System Configuration

The IDC consists of a high speed buffer memory, called a disk cache, and a controller integrated into the Input/Output Processor (IOP), as shown in Fig. 1.

The disk cache includes up to 16 megabytes of MOS random access storage which stores frequently used data under the IOP control. It is connected to the System Control Unit of the ACOS system and is shared by all the disks in the system. The data in the disk cache can be transferred to and from the main memory at a speed of up to 5.3 megabytes per second, which is three to six times faster than that of a conventional disk. Needless to say, seek time and latency time are not required to start the data transfer.

The IOP is an intelligent controller which handles all the input/output transactions issued by the central processing unit (CPU). This capability is used to control the IDC. For this purpose, a control program called Disk Cache

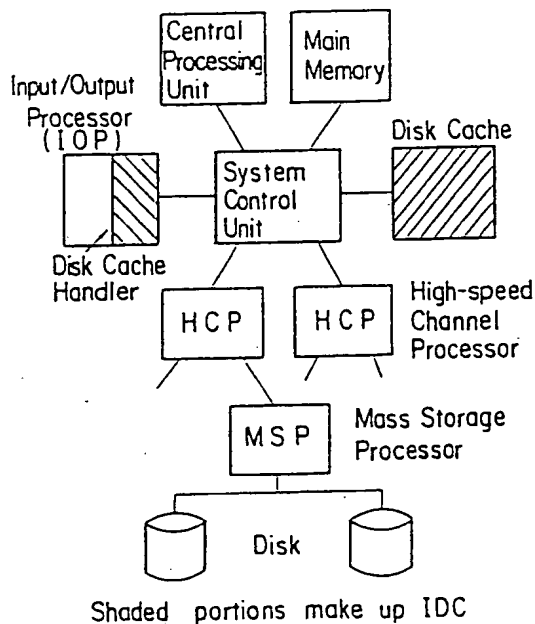


Fig. 1 IDC system configuration

handler is integrated into the main program of the IOP. The Disk Cache handler, interfacing with ACOS operating system, controls all the IDC operations, including directory management for locating the accessed data, data transfer between the disks and the disk cache, and data transfer between the disk cache and the main memory.

Sector Addressing and Data Mapping

Although the record read or written by the operating system is variable in length, data are recorded on a disk in fixed blocks, called sectors. A number of sectors make up the record. The sector has a unique address and all the accesses to the disk data are made using this address. This addressing allows effective data mapping and adaptive control by the IDC, independent of the physical structure of disk devices being used.

Like a CPU cache, the disk data are mapped into the disk cache using a set associative mapping, as shown in Fig. 2. The basic mapping unit is a block which consists of a number of sectors. The block size as well as the number of sets and the number of levels are adjustable to each user environment.

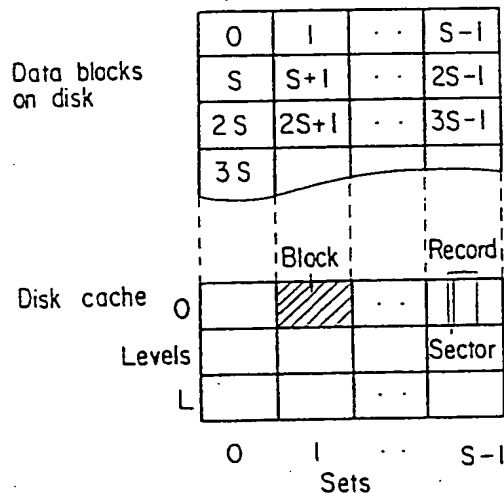


Fig. 2 Data mapping

Basic Operations

Before explaining the file adaptive control, basic cache operations are described here.

(1) Read Operation

If the accessed record is found in the disk cache, the record is transferred to the main memory. This completes the read operation. If the record is not found in the disk cache, the data block which contains that

record is read from a disk into the disk cache and the record is transferred to the main memory. If no free cache block is available, least recently used (LRU) cache block is replaced.

(2) Write Operation

The record is always written onto a disk. If an old copy of the record is found in the disk cache, the record is also transferred to the disk cache to update the copy. In either case, an I/O completion is sent to the operating system after the record has been written onto the disk. The disk access time, therefore, is not improved by the disk cache. However, since this operation guarantees that the data in the disk are always what the operating system assumes, no change is required in the error handling processing.

File Adaptive Control

Four Operation Modes

Four operation modes are provided in the IDC to realize high performance while assuring sufficient availability (Table 1). These are; (1) Basic mode, (2) Sequential File mode, (3) Temporary File mode and (4) High Speed File mode. Modes (1), (2) and (3) can be specified independently at each I/O request. Mode (4) can be specified on a file basis.

(1) Basic mode

This is the mode intended for a random file processing and performs the basic operations described in the previous section.

(2) Sequential File mode

The following two operations are added to the basic operations to utilize the nature of sequential file processing.

(a) When the record to be read is not found in the cache, disk blocks which contain n records are prefetched into the cache for subsequent use. This guarantees an average

hit ratio (a probability that a record is found in the cache) of $(n-1)/n$ for any sequential file processing. For example, assuming a record longer than a cache block, the cache hit ratio goes up from 0 % to 67 % by applying this control with $n=3$. Another merit of this control is that it allows the cache block size to be decreased or tuned to fit other file categories, irrespective of the sequential file performance.

(b) A cache block, whose last record has been accessed, is placed at the head of the LRU list for immediate use by other files. This is based on the fact that the same data is rarely accessed again soon in a sequential file processing. This control realizes efficient usage of cache space and an increased overall hit ratio. Without this control, one sequential file could consume all the cache blocks. However, with this control, the sequential file is confined to a few blocks within one level.

(3) Temporary File mode

A temporary file processing usually starts with writing a new record, followed by eventual reading of the record. Therefore, this mode is designed to operate in such a way that a record is always written into the disk cache, assigning a new block if necessary. This control increases the hit ratio in read operations. For example, assuming a work file processing involving writing a small amount of data and reading it at once, read hit ratio goes up from 0 % to 100 % by adding this control to the basic mode.

(4) High Speed File mode

Whereas all the modes described above improve only read access time, this mode is designed to decrease both read and write access times. One way to achieve this objective is to notify an I/O completion to the operating system, immediately after a record has been written into the disk cache. The record is written onto a disk

Table 1 Summary IDC operations

Operation modes	Read		Write	
	Hit	Miss	Hit	Miss
Basic	DC-MM-▽	DK⇒DC-MM-▽	MM{DC DK}▽	MM-DK-▽
Sequential		DK*DC-MM-▽		MM{DK DC}▽
Temporary File		DK⇒DC-MM-▽		
High Speed File		—	MM⇒DC-▽	—

DC: Disk Cache DK: Disk MM: Main Memory ▽: I/O completion

—: record transfer ⇒: block transfer *: n records

thereafter. This method, however, has the following problem. When a failure occurs and the record cannot be successfully written onto a disk, there is no way to notify this event to the operating system, which has already finished the I/O processing, assuming its successful completion. Allowing such a dual notification requires significant modifications in the error handling and recovery routines for the operating system. To avoid this problem, it was decided not to write a record onto a disk but instead to restrict the use of this mode to a temporary file only.

The High Speed File mode operates as if a portion of file space is allocated in the disk cache. A file to be handled in this mode must be registered in the IDC. In other words, the operating system first sends to the IDC its size and start address on disk, and sends a release command when the file processing finishes. When disk records in the registered file area are accessed, the records are allocated in the disk cache block by block. Once allocated, these blocks remain in the disk cache until a release command is received. When no cache space is available, the records are handled as a Temporary File mode and are written onto a disk. To prevent this mode from exploiting the cache, an upper capacity limit is provided for this mode.

Control Interface

Figure 3 shows the control interface

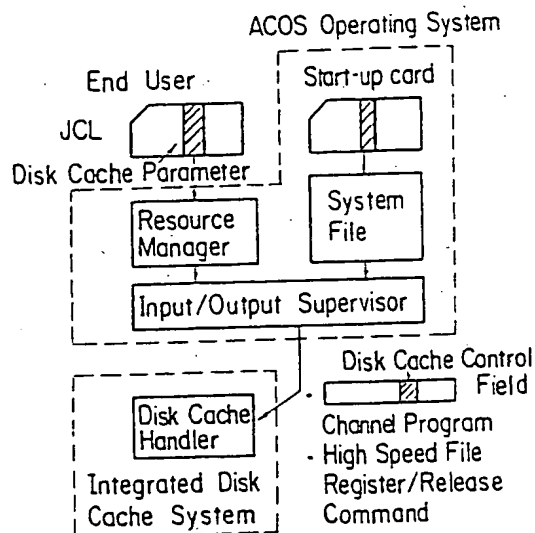


Fig. 3 IDC control interface

structure for the file adaptive control. For the interface between the operating system and the IDC, a new control field is provided in a channel program control word. In this field, the Input Output Supervisor specifies, for each disk I/O operation, IDC on/off or an operation mode to be used. The IDC, based on this control field, dynamically changes its operation mode and performs the appropriate operations. In the High Speed File mode, a newly provided file registration/release command is also used.

For a user interface, several system defaults are provided for easy IDC control. These include automatic IDC application to all sequential files, TSS swap files, system library file etc. An end user, if necessary, can also control the IDC use on a file basis. For this purpose, an IDC parameter was added to a job control card.

Error Recovery

The disk cache is designed to have sufficiently high reliability through its error correcting capability. If an uncorrectable error occurs, however, the associated block is removed from the system and the IDC continues its operation in a degraded mode. When data in the disk cache becomes inaccessible, due to some hardware failure, the I/O request involved in the failure is converted to a usual disk I/O operation. Except for the High Speed File mode, this conversion is always possible because the IDC is controlled in such a way that the data in the disk cache always has its copy on a disk. In the High Speed File mode, the I/O request involved in the failure ends in an I/O device error.

In summary, when an IDC failure occurs, the IDC is simply bypassed in either mode other than the High Speed File mode. In the latter mode, only a job involved in the failure is aborted.

Performance Gains

Access Time Improvement

Disk record read time (t), when using the IDC, is given by

$$t = p \cdot r + (1-p) (b + r) \\ = r + (1-p)b$$

where

- p : disk cache hit ratio
- r : record read time from the disk cache
- b : block(s) load time from a disk into the disk cache

Based on this formula, the average effective disk access time is calculated and compared with the conventional disk access time in Fig. 4.

As can be seen from this figure, an improvement factor of more than 10 is obtained by the IDC for reading the data in the disk cache or accessing the data in the High Speed File mode. In other cases, however, the data access time depends on the cache hit ratio and the percentage of read access occurrences. If 80 - 90 % hit ratio and 6 : 4 read/write ratio is assumed, as is often found to be the case,

BEST AVAILABLE COPY

the access time can improve by a factor of two for overall accesses and by three to six for read only accesses.

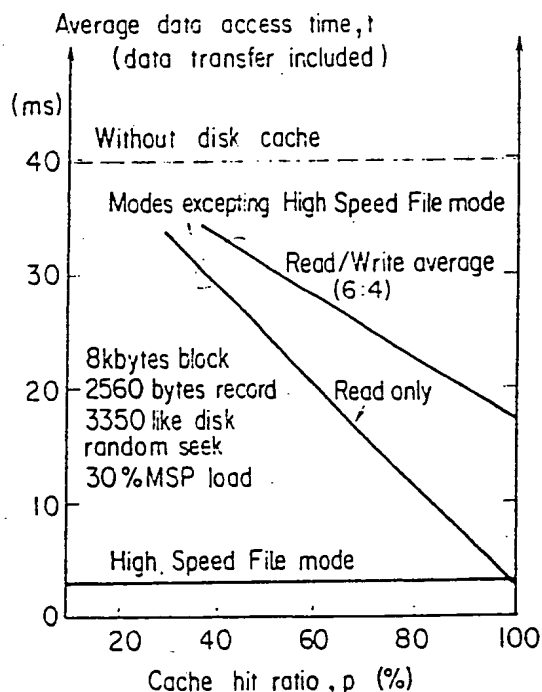


Fig. 4 Disk data access time with IDC

Sample Measurement

The improvement in throughput or response time, which is obtained by applying the IDC, depends on the application, system operation parameters and the IDC parameters among other factors. Therefore, the performance gains obtained by the IDC should be discussed separately for each user system. This is beyond the scope of this paper. Here, only sample IDC performance gains are shown in Table 2, which are based upon the measurement of user model jobs.

Application	Improvement	Hit ratio
Compilation	10-20%(elapsed time)	85-90%
COBOL	10-35 (")	80-95
SORT	10-30 (")	60-90
TSS	15-25 (response time)	75-95

3330 like disk Disk cache 0.5-6Mbytes

Note: High Speed File mode is not used.

Table 2 Sample IDC performance measurement

Conclusion

The Integrated Disk Cache system, which includes up to 16 megabytes of MOS memory, has been developed. From performance and reliability viewpoints, the IDC employs file adaptive control, in which four operation modes are provided and their use is specified by the operating system. The IDC improves the average disk access time by a factor of two to ten and user job performance by 10 to 35 %. File adaptive control, like the one presented here, is thought to be an effective method to realize a high performance disk cache system.

Acknowledgement

Many people have contributed to the development of the IDC. The authors especially wish to extend their appreciation to T. Kitamura, T. Inawashiro, S. Tanaka and T. Tashiro for their helpful suggestions and discussions.

References

- (1) A. J. Smith, "On the Effectiveness of Buffered and Multiple ARM Disks," Proc. Fifth Symposium on Computer Architecture, Apr. 1973, pp242-248.
- (2) T. A. Welch, "Analysis of Memory Hierarchies for Sequential Data Accesses," Computer, Vol 12, No. 5, May 1979, pp19-26.
- (3) 3770 Disc Cache, Memorex Product Description Manual, No. 3770-00, May 1978.